APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/662,087	09/12/2003	Liem Gioi Tran	SVL920030083US1/IBMP029	8862

63056	7590	11/28/2007

MOLLBORN PATENTS
ATTN: IBM
2840 COLBY DRIVE
BOULDER, CO 80305

EXAMINER
SMITH, GARRETT A

ART UNIT	PAPER NUMBER
2168	

NOTIFICATION DATE	DELIVERY MODE
11/28/2007	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

MOLLBORN@MOLLBORN.COM
sbailey@mollborn.com

	Application No.	Applicant(s)
Office Action Summary	10/662,087	TRAN, LIEM GIOI
	Examiner	Art Unit	
	Garrett A. Smith	2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on <u>21 September 2007</u>.

2a) ☒ This action is **FINAL**.   2b) ☐ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) <u>*1-20*</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) <u>*1-20*</u> is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1.   This Office action is regarding Applicant's response filed 21 September 2007 to a

prior Office action. Claims 1 – 20 are pending. Claims 7 – 14 and 20 are amended.

Response to Arguments

35 USC 101

2.   Applicant's arguments (page 6) and amendments, filed 21 September 2007,

regarding the rejection under 35 USC 101 of claims 8 – 14 have been fully considered

and are persuasive. For these reasons, the rejection under 35 USC 101 of claims 8 –

14 is **<u>withdrawn</u>**.

3.   Applicant's arguments (page 6), filed 21 September 2007, regarding the rejection

under 35 USC 101 of claims 15 – 20 have been fully considered but they are not

persuasive.

Applicant argues, "The Applicant respectfully believes that these claims are

directed to statutory matter and that the rejection under 35 U.S.C § 101 of claims 15 –

20 was made in error by the Examiner, especially since the Examiner does not address

these claims in the rejection." The Examiner respectfully points out that claims 15 – 20

were addressed both in the rejection heading and the body of the rejection. Further, in

claims 15 – 20, each of the components or "means" described are, in fact, directed to

software alone. Each "means" can be implemented wholly in software.   In order to be

considered statutory under 35 USC 101, each of the embodiments covered by the

claims must also be statutory.

The Examiner would like to further point out that claims 15 – 20 **do not** invoke 35 USC 112 6th Paragraph. 35 USC 112 6th Paragraph states,

> An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.

As software code is not structure, material or acts, the claims cannot meet the requirements of 35 USC 112 6th Paragraph. While software code can provide instructions to perform acts, software code, in itself, is not an act. Software code is neither a structure nor material.

For these reasons, the rejection under 35 USC 101 of claim 15 – 20 is **maintained**.

35 USC 103(a)

4.   Applicant's arguments (page 7), filed 21 September 2007, regarding the rejection under 35 USC 103(a) of claims 1 – 20 have been fully considered but they are not persuasive. Applicant argues that a person of ordinary skill in the art at the time of invention lacks motivation to combine Zeller et al with SQL-92. The Examiner notes that according to *Teleflex Inc. v. KSR Int'l Co.*, 550 U.S.– 82 USPQ2d 1385 (2007) a motivation to combine is not required. Further, showing a "resulting improvement" is also not required. The Examiner must show that these references are readily combinable by a person of ordinary skill in the art to yield a predictable result. As Zeller et al is directed to a SQL query normalizer (and related processes) and SQL-92 is the basic specification on which the SQL language is built, these are clearly in the same art. These references are trivially combined as Zeller et al uses SQL in the application of the

normalizer and SQL-92 is the basis of the language used. Therefore, while Zeller has

all of the functions, it does not explicitly state a cast function function. SQL-92, on the

other hand, has explicitly a cast function. As to the actual combination of the two

references and the reasonable expectation of success, the Examiner points out that the

claims only require that a SQL template be "converted" by a cast function and storing

the resulting data type. The SQL template is not limited to a particular size and

therefore can be reasonably be a single expression with a single item. As typecasting a

single item is quite clearly shown in SQL-92 (section 6.10), it would be reasonable to

expect success from the combination. For these reasons, the rejection under 35 USC

103(a) of claims 1 – 20 is **maintained**.

Claim Rejections - 35 USC § 101

5.   35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of
> matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the
> conditions and requirements of this title.

6.   Claims **15 – 20** are rejected under 35 U.S.C. 101 because the claimed invention

is directed to non-statutory subject matter.

7.   **Claims 15 – 20** are directed towards software, per se. The claims lack the

necessary physical articles or objects to constitute a machine or a manufacture within

the meaning of 35 USC 101. They are clearly not a series of steps or acts to be a

process nor are they a combination of chemical compounds to be a composition of

matter. As such, they fail to fall within a statutory category. They are, at best, functional

descriptive material *per se*. Descriptive material can be characterized as either

"functional descriptive material" or "nonfunctional descriptive material." Both types of

"descriptive material" are nonstatutory when claimed as descriptive material *per se*, 33

F.3d at 1360, 31 USPQ2d at 1759. When <u>functional</u> descriptive material is recorded on

some computer-readable medium, it becomes structurally and functionally interrelated

to the medium and will be statutory in most cases since use of technology permits the

function of the descriptive material to be realized. Compare *In re Lowry*, 32 F.3d 1579,

1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994). Merely claiming <u>nonfunctional</u>

descriptive material, i.e., abstract ideas, stored on a computer-readable medium, in a

computer, or on an electromagnetic carrier signal, does not make it statutory. See

Diehr, 450 U.S. at 185-86, 209 USPQ at 8 (noting that the claims for an algorithm in

Benson were unpatentable as abstract ideas because "[t]he sole practical application of

the algorithm was in connection with the programming of a general purpose

computer.").

Claim Rejections - 35 USC § 103

8.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

9.   Claims **1 – 20** are rejected under 35 U.S.C. 103(a) as being unpatentable over

Zeller et al (US Patent 5,724,570, dated 3 March 1998) and further in view of SQL-92

Specification section 6.10 (hereinafter SQL-92).

10.   In regard to **claim 1**, Zeller et al teaches converting the SQL template into a

converted SQL template (*the SQL query is converted by a normalizer, see col 7, lines 1*

– 17); acquiring a data type of the converted SQL template (*the data type is known as*

the query is acted on, such as BOOLEAN, see col 8, lines 27 – 31); and storing the data

type of the converted SQL template with the SQL template (*the data type and the can*

be stored in memory or on hard disk F100, see col 7, lines 1 – 17). However, Zeller et

al does not teach the explicit use of a cast function. SQL-92 does teach that a cast

function can be used to change or recast an item as a different data type. It would have

been obvious to a person of ordinary skill in the art to use the method of Zeller et al with

the cast function of SQL-92 because the cast function allows for queries with non-

homogeneous data types to be operated on without data type errors.

11.   In regard to **claim 2**, Zeller et al further teaches converting the SQL template

comprises replacing tokens in the SQL template (*See Fig 3A and 3B as well as col 11,*

lines 21 – 30; the nested queries are replaced).). However, Zeller et al does not teach

the explicit use of a cast function. SQL-92 does teach that a cast function can be used

to change or recast an item as a different data type. It would have been obvious to a

person of ordinary skill in the art to use the method of Zeller et al with the cast function

of SQL-92 because the cast function allows for queries with non-homogeneous data

types to be operated on without data type errors.

12.   In regard to **claim 3**, Zeller et al further teaches forming a valid SQL statement

from the converted SQL template (*from an optimized query tree, equivalent SQL*

statement can be formed and executed by the executor module F110, see col 7, lines 1

– 17). SQL-92 does teach that a cast function can be used to change or recast an item as a different data type. It would have been obvious to a person of ordinary skill in the art to use the method of Zeller et al with the cast function of SQL-92 because the cast function allows for queries with non-homogeneous data types to be operated on without data type errors.

13.   In regard to **claim 4**, Zeller et al does not teach the explicit use of a cast function to determine validity of SQL statement. However, as shown by SQL-92, validity of a query can be determined by the results of a cast function. It would have been obvious to a person of ordinary skill in the art to use the method of Zeller et al with the cast function of SQL-92 because the cast function allows for queries with non-homogeneous data types to be operated on without data type errors.

14.   In regard to **claim 5**, Zeller et al further teaches acquiring the data type of the converted SQL statement comprises passing the valid SQL statement through an SQL processor (*the data type is known as the query is acted on, such as BOOLEAN, see col 8, lines 27 – 31; the query is acted on by the optimizer with is equivalent to a SQL processor*). SQL-92 does teach that a cast function can be used to change or recast an item as a different data type. It would have been obvious to a person of ordinary skill in the art to use the method of Zeller et al with the cast function of SQL-92 because the cast function allows for queries with non-homogeneous data types to be operated on without data type errors.

15.   In regard to **claim 6**, Zeller et al further teaches inquiring if a descendent of the converted SQL template has been modified (*the optimizer and normalizer does*

operations based on rules which will transverse the tree and check for modifications, col 7, lines 1 – 17). It would have been obvious to a person of ordinary skill in the art to use the method of Zeller et al with the cast function of SQL-92 because the cast function allows for queries with non-homogeneous data types to be operated on without data type errors.

16.   In regard to **claim 7**, Zeller et al further teaches if the descendent of the converted SQL template has been modified, re-evaluating an SQL template for the descendent and cascading a modified data type up to ancestors of the converted SQL template (*the optimizer and normalizer does operations based on rules which will transverse the tree and check for modifications, col 7, lines 1 – 17, data types are moved up to the parent query as necessary for computation*). It would have been obvious to a person of ordinary skill in the art to use the method of Zeller et al with the cast function of SQL-92 because the cast function allows for queries with non-homogeneous data types to be operated on without data type errors.

17.   In regard to **claim 8**, Zeller et al teaches converting the SQL template into a converted SQL template (*the SQL query is converted by a normalizer, see col 7, lines 1 – 17*); acquiring a data type of the converted SQL template (*the data type is known as the query is acted on, such as BOOLEAN, see col 8, lines 27 – 31*); and storing the data type of the converted SQL template with the SQL template (*the data type and the can be stored in memory or on hard disk F100, see col 7, lines 1 – 17*). However, Zeller et al does not teach the explicit use of a cast function. SQL-92 does teach that a cast function can be used to change or recast an item as a different data type. It would have

been obvious to a person of ordinary skill in the art to use the computer program

product of Zeller et al with the cast function of SQL-92 because the cast function allows

for queries with non-homogeneous data types to be operated on without data type

errors.

18.   In regard to **claim 9**, Zeller et al further teaches converting the SQL template

comprises replacing tokens in the SQL template (*See Fig 3A and 3B as well as col 11,*

lines 21 – 30; the nested queries are replaced).). However, Zeller et al does not teach

the explicit use of a cast function. SQL-92 does teach that a cast function can be used

to change or recast an item as a different data type. It would have been obvious to a

person of ordinary skill in the art to use the computer program product of Zeller et al

with the cast function of SQL-92 because the cast function allows for queries with non-

homogeneous data types to be operated on without data type errors.

19.   In regard to **claim 10**, Zeller et al further teaches forming a valid SQL statement

from the converted SQL template (*from an optimized query tree, equivalent SQL*

statement can be formed and executed by the executor module F110, see col 7, lines 1

– 17). SQL-92 does teach that a cast function can be used to change or recast an item

as a different data type. It would have been obvious to a person of ordinary skill in the

art to use the computer program product of Zeller et al with the cast function of SQL-92

because the cast function allows for queries with non-homogeneous data types to be

operated on without data type errors.

20.   In regard to **claim 11**, Zeller et al does not teach the explicit use of a cast

function to determine validity of SQL statement. However, as shown by SQL-92, validity

of a query can be determined by the results of a cast function. It would have been

obvious to a person of ordinary skill in the art to use the computer program product of

Zeller et al with the cast function of SQL-92 because the cast function allows for queries

with non-homogeneous data types to be operated on without data type errors.

21.   In regard to **claim 12**, Zeller et al further teaches acquiring the data type of the

converted SQL statement comprises passing the valid SQL statement through an SQL

processor (*the data type is known as the query is acted on, such as BOOLEAN, see col*

8, lines 27 – 31; the query is acted on by the optimizer with is equivalent to a SQL

processor). SQL-92 does teach that a cast function can be used to change or recast an

item as a different data type. It would have been obvious to a person of ordinary skill in

the art to use the computer program product of Zeller et al with the cast function of SQL-

92 because the cast function allows for queries with non-homogeneous data types to be

operated on without data type errors.

22.   In regard to **claim 13**, Zeller et al further teaches inquiring if a descendent of the

converted SQL template has been modified (*the optimizer and normalizer does*

operations based on rules which will transverse the tree and check for modifications, col

7, lines 1 – 17). It would have been obvious to a person of ordinary skill in the art to use

the computer program product of Zeller et al with the cast function of SQL-92 because

the cast function allows for queries with non-homogeneous data types to be operated

on without data type errors.

23.   In regard to **claim 14**, Zeller et al further teaches if the descendent of the

converted SQL template has been modified, re-evaluating an SQL template for the

descendent and cascading a modified data type up to ancestors of the converted SQL

template (*the optimizer and normalizer does operations based on rules which will*

transverse the tree and check for modifications, col 7, lines 1 – 17, data types are

moved up to the parent query as necessary for computation). It would have been

obvious to a person of ordinary skill in the art to use the computer program product of

Zeller et al with the cast function of SQL-92 because the cast function allows for queries

with non-homogeneous data types to be operated on without data type errors.

24.   In regard to **claim 15**, Zeller et al teaches converting the SQL template into a

converted SQL template (*the SQL query is converted by a normalizer, see col 7, lines 1*

– 17); acquiring a data type of the converted SQL template (*the data type is known as*

the query is acted on, such as BOOLEAN, see col 8, lines 27 – 31); and storing the data

type of the converted SQL template with the SQL template (*the data type and the can*

be stored in memory or on hard disk F100, see col 7, lines 1 – 17). However, Zeller et

al does not teach the explicit use of a cast function. SQL-92 does teach that a cast

function can be used to change or recast an item as a different data type. It would have

been obvious to a person of ordinary skill in the art to use the system of Zeller et al with

the cast function of SQL-92 because the cast function allows for queries with non-

homogeneous data types to be operated on without data type errors.

25.   In regard to **claim 16**, Zeller et al further teaches converting the SQL template

comprises replacing tokens in the SQL template (*See Fig 3A and 3B as well as col 11,*

lines 21 – 30; the nested queries are replaced).). However, Zeller et al does not teach

the explicit use of a cast function. SQL-92 does teach that a cast function can be used

to change or recast an item as a different data type. It would have been obvious to a person of ordinary skill in the art to use the system of Zeller et al with the cast function of SQL-92 because the cast function allows for queries with non-homogeneous data types to be operated on without data type errors.

26.   In regard to **claim 17**, Zeller et al further teaches forming a valid SQL statement from the converted SQL template (*from an optimized query tree, equivalent SQL statement can be formed and executed by the executor module F110, see col 7, lines 1 – 17*). SQL-92 does teach that a cast function can be used to change or recast an item as a different data type. It would have been obvious to a person of ordinary skill in the art to use the system of Zeller et al with the cast function of SQL-92 because the cast function allows for queries with non-homogeneous data types to be operated on without data type errors.

27.   In regard to **claim 18**, Zeller et al does not teach the explicit use of a cast function to determine validity of SQL statement. However, as shown by SQL-92, validity of a query can be determined by the results of a cast function. It would have been obvious to a person of ordinary skill in the art to use the system of Zeller et al with the cast function of SQL-92 because the cast function allows for queries with non-homogeneous data types to be operated on without data type errors.

28.   In regard to **claim 19**, Zeller et al further teaches acquiring the data type of the converted SQL statement comprises passing the valid SQL statement through an SQL processor (*the data type is known as the query is acted on, such as BOOLEAN, see col 8, lines 27 – 31; the query is acted on by the optimizer with is equivalent to a SQL*

processor). SQL-92 does teach that a cast function can be used to change or recast an item as a different data type. It would have been obvious to a person of ordinary skill in the art to use the system of Zeller et al with the cast function of SQL-92 because the cast function allows for queries with non-homogeneous data types to be operated on without data type errors.

29.   In regard to **claim 20**, Zeller et al further teaches if the descendent of the converted SQL template has been modified, re-evaluating an SQL template for the descendent and cascading a modified data type up to ancestors of the converted SQL template (*the optimizer and normalizer does operations based on rules which will transverse the tree and check for modifications, col 7, lines 1 – 17, data types are moved up to the parent query as necessary for computation*). It would have been obvious to a person of ordinary skill in the art to use the system of Zeller et al with the cast function of SQL-92 because the cast function allows for queries with non-homogeneous data types to be operated on without data type errors.

Conclusion

30.   **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time

policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not

mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later

than SIX MONTHS from the mailing date of this final action.

31.   The Examiner requests, in response to this Office action, that support be shown

for language added to any original claims on amendment and any new claims. That is,

indicate support for newly added claim language by specifically pointing to page(s) and

line no(s) in the specification and/or drawing figure(s). This will assist the Examiner in

prosecuting the application.

32.   When responding to this Office action, Applicant is advised to clearly point out

the patentable novelty which he or she thinks the claims present, in view of the state of

the art disclosed by the references cited or the objections made. He or she must also

show how the amendments avoid such references or objections See 37 CFR 1.111(c).

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Garrett A. Smith whose telephone number is (571) 270-

1764. If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Tim T. Vo can be reached on (571) 272-3642. The fax phone number for

the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a

USPTO Customer Service Representative or access to the automated information

system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

November 14, 2007   Garrett Smith
                                                       Patent Examiner
                                                       Art Unit 2168

                                                       TIM VO
                                                SUPERVISORY PATENT EXAMINER
                                                 TECHNOLOGY CENTER 2100